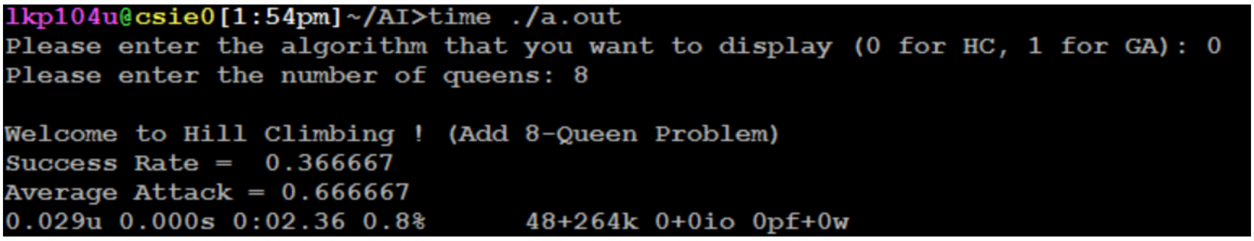
AI 人工智慧　　　　404410077 林楷博

1. For the 8-queen problem (n = 8): (時間為執行 30 次的結果)

Q1: List all the results (average #attacks in the final configuration) from the two methods.

Ans:

(1) HC:

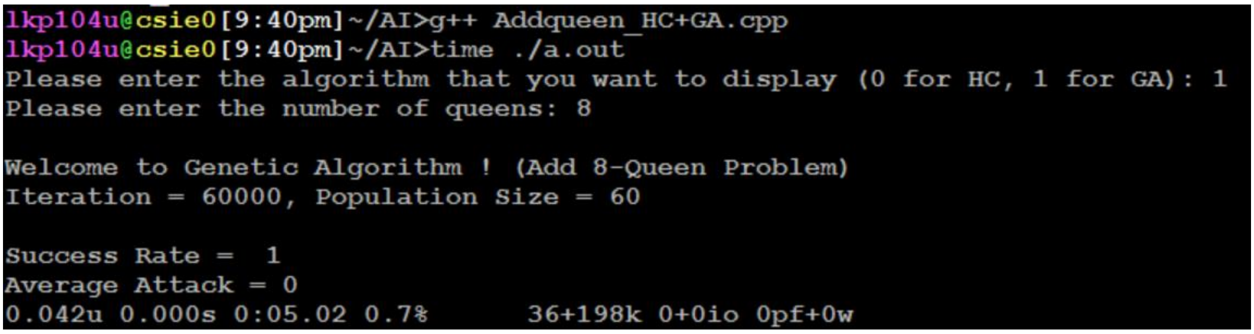


```
lkp104u@csie0[1:54pm]~/AI>time ./a.out
Please enter the algorithm that you want to display (0 for HC, 1 for GA): 0
Please enter the number of queens: 8

Welcome to Hill Climbing ! (Add 8-Queen Problem)
Success Rate =  0.366667
Average Attack = 0.666667
0.029u 0.000s 0:02.36 0.8%      48+264k 0+0io 0pf+0w
```

Average Attack = 0.666667

(2) GA (Iteration = 60000, Population Size = 60):



```
lkp104u@csie0[9:40pm]~/AI>g++ Addqueen_HC+GA.cpp
lkp104u@csie0[9:40pm]~/AI>time ./a.out
Please enter the algorithm that you want to display (0 for HC, 1 for GA): 1
Please enter the number of queens: 8

Welcome to Genetic Algorithm ! (Add 8-Queen Problem)
Iteration = 60000, Population Size = 60

Success Rate =  1
Average Attack = 0
0.042u 0.000s 0:05.02 0.7%      36+198k 0+0io 0pf+0w
```

Average Attack = 0

(3) Compare → GA (Success Rate: Always 1) always gets the best answer at 30 runs in the 8-queen problem, while HC (Success Rate: 0.2~0.5) doesn't. At this case, GA gets the optimal solution every time, means that it never attack at any round. On the other hand, HC average attacks two thirds every round.

Q2: Compare the average running time for the two methods to get a solution.

Ans:

(1) HC: (圖為執行 30 次的時間)

Running Time = CPU Time = User time + System Time

= (0.029 + 0.000) / 30

約 0.001166667 (sec)

(2) GA (Iteration = 60000, Population Size = 60):

Running Time = CPU Time = User time + System Time

= (0.042 + 0.000) / 30

約 0.0014 (sec)

(3) Compare → GA is about 1.2 times slower than HC (due to the Iterations = 60000 and Population Size = 60).

Q3: Compare the success rate (SR) of HC and GA.

Ans:

(1) HC:

Success Rate = 0.366667

(2) GA (Iteration = 60000, Population Size = 60):

Success Rate = 1

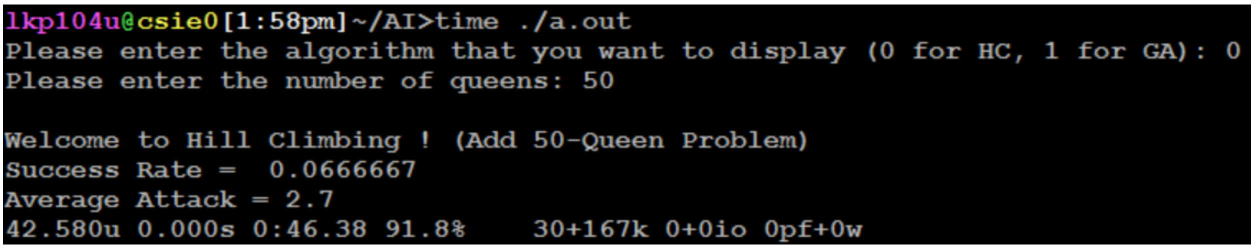
(3) Compare → GA (Success Rate: Always 1) always gets the optimal solution at the 8-queen problem every time, however HC (Success Rate: 0.2~0.5) performs lower. In this case GA perfectly succeed every time, while HC was only about 36.67%.

2. What if the 50-queen problem (n = 50) provided a 50×50 chessboard?

Q1: List all the results (average #attacks in the final configuration) from the two methods. (時間為執行 30 次的結果)

Ans:

(1) HC:

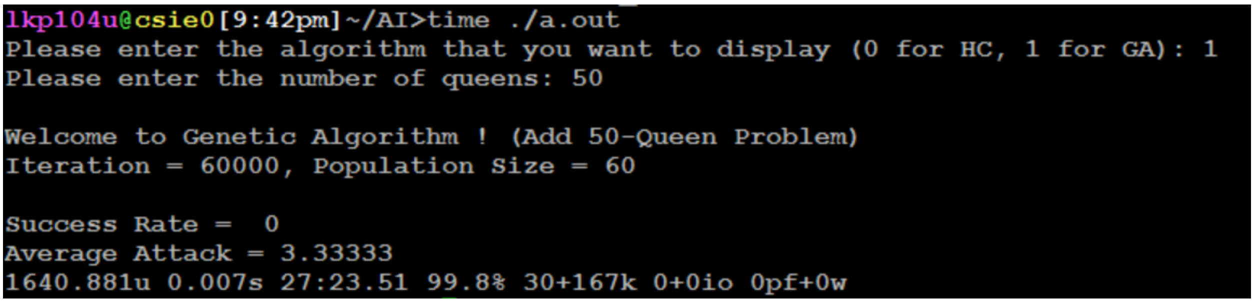


```
lkp104u@csie0[1:58pm]~/AI>time ./a.out
Please enter the algorithm that you want to display (0 for HC, 1 for GA): 0
Please enter the number of queens: 50

Welcome to Hill Climbing ! (Add 50-Queen Problem)
Success Rate =  0.0666667
Average Attack = 2.7
42.580u 0.000s 0:46.38 91.8%    30+167k 0+0io 0pf+0w
```

Average Attack = 2.7

(2) GA (Iteration = 60000, Population Size = 60):



```
lkp104u@csie0[9:42pm]~/AI>time ./a.out
Please enter the algorithm that you want to display (0 for HC, 1 for GA): 1
Please enter the number of queens: 50

Welcome to Genetic Algorithm ! (Add 50-Queen Problem)
Iteration = 60000, Population Size = 60

Success Rate =  0
Average Attack = 3.33333
1640.881u 0.007s 27:23.51 99.8% 30+167k 0+0io 0pf+0w
```

Average Attack = 3.33333

(3) Compare → HC's average attack performs better than GA's.

Q2: Compare the average running time for the two methods to get a solution.

Ans:

(1) HC: (圖為執行 30 次的時間)

Running Time (一次) = CPU Time = User time + System Time

= (42.580 + 0.000) / 30

約 1.49333333 (sec)

(2) GA (Iteration = 60000, Population Size = 60):

Running Time = (一次) = CPU Time = User time + System Time

= (1640.881 + 0.000) / 30

約 54.6960333 (sec)

(3) Compare → GA is about 36.5 times slower than HC. Due to the searching space is bigger (takes more time) and number of iteration and population size is still. Executing GA takes far more time than performing HC.

Q3: Compare the success rate (SR) of HC and GA.

Ans:

(1) HC:

Success Rate = 0.0666667

(2) GA (Iteration = 60000, Population Size = 60):

Success Rate = 0

(3) Compare → In this case, GA finds 0 optimal solutions in the 50-queen problem while HC finds twice. We find out that HC performs a little bit better than GA in the 50-queen problem. The possible reason why GA didn't perform better is because the iteration and the population size isn't large enough.

3. Must describe your methods and list their parameter setting for the experiments.

Ans:

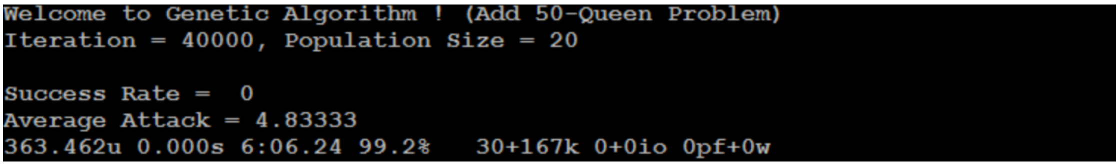
(1) HC: 先一個 row 一個 row 的產生 map，再利用 position[row]來記錄每一個 Queen(總數為 Q)的所在位置。由於 Queen 的數量是使用者自訂的，所以二維的 map 陣列必須利用 pointer 傳遞，再依據使用者所輸入的 Q 的大小，用 new 來建構出來(這邊編譯一直沒過，我卡超久的......)。更新好之後，再利用 attack 來計算出每一個 Queen 移動過後所會產生的 attack 數(如課本 p.123 的那張圖)，最後再用 Hill 函式，去移動到比 min 還小或是等於的點。反覆跑 attack 和 Hill 44 次，直到完整的跑完或是 min==0 為止。

(2) GA: 我的 GA 的 Crossover 是採用投影片 p.33 的 Order Crossover，所有

的資料都以 data(一維陣列做儲存)，而 Mutation 則是採用 p.37 的 Swap Mutation，藉由 random 隨機的將兩兩陣列元素交換。這次遇到最大的困難就是 Crossover 的實作，變數的設定和使用常常搞得我頭昏腦脹的，後來經過詢問同學和上網查找資料，才慢慢的把它建構出來。

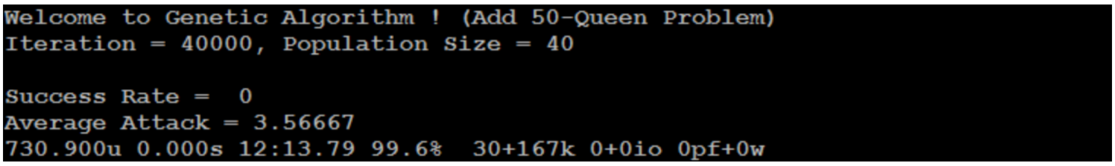
在跑族群之前，我會先將它們隨機抽樣產生不同的 data，再經由我的 attack2 function，將每筆資料的 attack 數都分別累加起來，最後經由排序相互比大小，從中取出一半的資料，做為下一個親代。最後經過慢慢的演化之後，就可以得到 GA 的 local 甚至是 global optimal solution 了。

另外可能是因為 Iteration 和 Population Size 不夠大的關係，我在跑 GA 的 50 queens 時，Success Rate 為 0。但由於 GA 的迴圈次數過多，像我設在 Iteration = 60000, Population Size = 60 就要跑 30min 左右。因此，我想說就盡量讓 GA 的 Iteration 和 Population Size 設在一個合理的範圍內，當作是有無找到 Optimal Solution 結果。



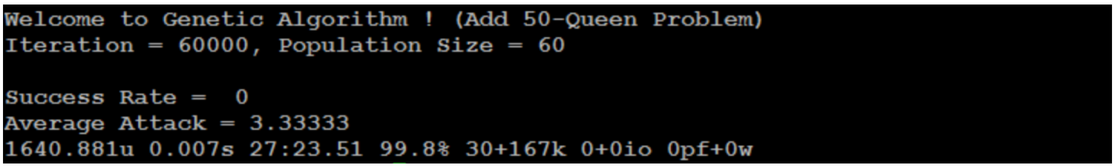
```
Welcome to Genetic Algorithm ! (Add 50-Queen Problem)
Iteration = 40000, Population Size = 20

Success Rate =  0
Average Attack = 4.83333
363.462u 0.000s 6:06.24 99.2%   30+167k 0+0io 0pf+0w
```



```
Welcome to Genetic Algorithm ! (Add 50-Queen Problem)
Iteration = 40000, Population Size = 40

Success Rate =  0
Average Attack = 3.56667
730.900u 0.000s 12:13.79 99.6%  30+167k 0+0io 0pf+0w
```



```
Welcome to Genetic Algorithm ! (Add 50-Queen Problem)
Iteration = 60000, Population Size = 60

Success Rate =  0
Average Attack = 3.33333
1640.881u 0.007s 27:23.51 99.8% 30+167k 0+0io 0pf+0w
```

(增加 Iteration 和 Population Size 會使 Average Attack 降低，但仍然不會 Success)